Amendments to the Specification:

Please replace the paragraph beginning at page 2, line 3 with the following amended paragraph:

To minimize the effects of misregistration, a technique known as trapping is used to adjust the shapes of color regions by spreading (expanding) some color regions to prevent gaps, and choking (contracting) other color regions to prevent overlaps. In determining whether an edge of a color region requires a trap, trapping systems evaluate the color on both sides of the edge. A method and apparatus for electronically trapping a printed color page in a desktop publishing, layout, graphics, or similar applications program is set forth in U.S. Patent No. ~~5,295,235~~ 5,295,236, entitled "Applying Traps to a Printed Page Specified in a Page Description Language Format," assigned to the assignee of the present invention and hereby incorporated in its entirety. A method and apparatus for vector map representation of page information to perform trapping is set forth in U.S. Patent No. 6,031,544, entitled "Vector Map Planarization and Trapping," also assigned to the assignee of the present invention and hereby incorporated in its entirety.

Please replace the paragraph beginning at page 2, line 15 with the following amended paragraph:

When trapping a vector object to another vector object, the color on both sides of the edge separating the vector objects can be determined by planarizing the vector objects. A vector object may alternatively intersect a raster image that is made up by a plurality of image pixels. If a vector object intersects a raster image, the edge between the vector object and raster image (vector-image edge) will have the vector color on one side and image color on the other ~~side The~~ side. The actual colors along the vector-image edge can be determined by rendering the raster image and vector object into a pixel buffer at device resolution and using raster based edge detection to accurately determine the edges and colors. While this method guarantees high fidelity vector-image edge color determination, it has an associated computational overhead of rasterizing the data at device resolution and requires extra memory for a raster buffer. Furthermore, this method is likely to produce edges that are jagged (also referred to as

pixilation), placing additional complexity on the trapping process to either smooth the jagged edge or handle the extra data points to accommodate the pixilation.

Please replace the paragraph beginning at page 7, line 4 with the following amended paragraph:

Referring now to FIG. 1 and to FIG. 4, a process (100) for identifying colors along an edge between a vector object and an image in accordance with the invention will be explained. As shown in FIG. 4, a vector object (415) partly overlaps an image (410). The process first maps an edge (420) between the image (410) and the vector object (415) to a device space (400) (step 105). The process can use any suitable line drawing algorithm to identify each device pixel (405) intersected by edge (420) as a vector boundary pixel (step 110). Next, the process proceeds with identifying a set of image boundary pixels corresponding to the vector boundary pixels. A corresponding image boundary pixel is a device pixel (405) located on the image side of the edge (420) and adjacent to the vector boundary pixel. The process identifies the image boundary pixels by selecting one of the identified vector boundary pixels (step 115) and creating a vector, with a magnitude of one device space pixel (405), pointing in the direction of the image (410) relative to the edge (420) (step 120). ~~The vector can may be~~ The vector can be normal to the edge (420). Alternatively, the vector can be normal to an axis in device space that forms the smallest angle with the edge. When the vector has been created, the process applies the vector to a vector boundary pixel, using vector addition, to identify a corresponding image boundary pixel (step 125). The process then determines the color of the image boundary pixel by applying a centerscan rule to the image boundary pixel (step 130). In accordance with the centerscan rule, the process maps the center of the image boundary pixel in the device space (400) to an image pixel (410) in the image space. The process then assigns the color of the image pixel (410) to the image boundary pixel in the device space (400) (step 135). After the color has been assigned, the process checks if there are any more vector boundary pixels to process (step 140). If there are more vector boundary pixels to process, the process returns to step 115 and repeats steps 115 through 140 for each remaining vector boundary pixel until all vector pixels have been processed and the process ends.

Please replace the paragraph beginning at page 9, line 18 with the following amended paragraph:

In the second case, the image has been clipped, using a clipping path, in order to show the part of the vector object that was concealed in the previous case. The process will therefore be applied to the slanted edge (420) instead of the vertical edge (425). The process identifies a set of device ~~pixels that are is~~ pixels that are intersected by the edge ~~(425)~~ (420), that is, the clipping path. After identifying the set of device pixels, the process selects a device pixel in the set (step 215) and checks if the center of the selected device pixel (405) maps to the image (410) (step 220). The center of the selected device pixel is considered to map to the image (410) if the center falls within the original image, regardless of whether the center falls within the clipping path or not. If the clipping path is larger than the original image, the center of a device pixel may fall within the clipping path, but outside the original image. In this case, the device pixel is not considered to map to the image (410). In other words, the point of reference is always the original image, regardless of the size and shape of the clipping path as far as color mapping goes. The purpose of the clipping path is merely to define the region that is to be painted by the image pixel colors, that is, if the vertical edge (425) or the slanted edge (420) shown in FIG. 4 should be examined.

Please replace the paragraph beginning at page 10, line 20 with the following amended paragraph:

Referring to FIG. 3 and FIG. 5, a process (300) determines the color of Image B (510) along the edge (520) of Image A (515) and Image B (510), when image A is rendered after ~~image A~~ image B, that is, when image A has a higher paint order. The process first maps the edge (520) to device space (500) (step 305) and identifies a set of device pixels (505) intersected by the edge (520) (step 310). The process proceeds by selecting a device pixel in the set (step 315) and checks whether the device pixel is an Image A boundary pixel (or, alternatively, an Image B boundary pixel) (step 320). If the center of the device pixel (505) does not map to Image A, the process identifies the device pixel as an Image B boundary pixel (step 325) and continues to step 345 which will be described below. The mapping of the device pixel centers are carried out in a corresponding way to the mapping that was explained above with reference to

the example of a centerscan object having a higher paint order than the overscan object. Clipping paths are also treated in a corresponding manner. If the center of the device pixel (505) maps to Image A (510), the device pixel (505) is identified as an Image A boundary pixel (step 330) and the process identifies the corresponding Image B boundary pixel by creating a vector, specified in device space pixels, pointing in the direction of Image B (510) relative to the edge (520) (step 335). The vector can be normal to the edge (520). Alternatively, the vector can be normal to an axis in device space that forms the smallest angle with the edge. When the vector has been created, the process applies the vector to the Image A boundary pixel, using vector addition to identify a corresponding Image B boundary pixel (step 340). The process then determines the color of the Image B boundary pixel by applying a centerscan rule to the Image B boundary pixel (steps 345 and 350). After the color has been assigned, the process checks if there are any more device pixels intersected by the edge to process (step 355). If there are more device pixels to process, the process returns to step 315 and repeats steps 315 through 350 for each remaining device pixel until all the device pixels in the set have been processed and the process ends.

Please delete the paragraph beginning at page number 12, line number 31, which starts with "What is claimed is:"

Please add the following new paragraph beginning at page number 13, line number 1:

What is claimed is: